

## EXTANT CLAIMS

Following are clean copies of the extant claims, including any further amendment requested herewith:

1. A method for rendering 3D graphics, comprising the steps of:  
    rendering primitives which comprise a graphics display;  
    invoking suspension of reads of pixel information during rendering of primitives only selectively; and  
    writing data generated by said rendering step into a frame buffer, and outputting data from said frame buffer at video rates.
2. The method of Claim 1, wherein said rendering is performed in a rendering system which includes at least four functionally distinct processing units, allocated to individual rendering operations, connected in a pipeline relationship.
3. The method of Claim 1, wherein a table of pixel information is used to determine if reads should be suspended.
4. The method of Claim 1, wherein reads are suspended if a primitive being rendered will affect a pixel location which has been previously touched by the rendering of a primitive and has not yet been updated.
5. The method of Claim 1, wherein a table of pixel information to determine if reads should be suspended, said table comprising a first bit and a second bit and wherein said first bit flags all pixels whose information has not yet been updated or discarded and said second bit flags pixels which will be affected by said primitive to be rendered.
6. The method of Claim 1, wherein said invoking step is not performed if a primitive to be rendered appears in a history list of recently rendered primitive data.

7. A method for rendering 3D graphics, comprising the steps of:
- providing data to be rendered to a rendering system which includes at least four functionally distinct processing units connected in a pipeline relationship;
  - allocating individual rendering operations to respective ones of said processing units, and performing said rendering operations;
  - invoking suspension of reads of pixel information during rendering of primitives only selectively; and
  - writing data generated by said processing units into a frame buffer, and outputting data from said frame buffer at video rates.
8. The method of Claim 7, wherein a table of pixel information is used to determine if reads should be suspended.
9. The method of Claim 7, wherein reads are suspended if a primitive being rendered will affect a pixel location which has been previously touched by the rendering of a primitive and has not yet been updated.
10. The method of Claim 7, wherein a table of pixel information to determine if reads should be suspended, said table comprising a first bit and a second bit and wherein said first bit flags all pixels whose information has not yet been updated or discarded and said second bit flags pixels which will be affected by said primitive to be rendered.
11. The method of Claim 7, wherein said invoking step is not performed if a primitive to be rendered appears in a history list of recently rendered primitive data.

12. A method of computer graphics memory management, comprising the actions of:

tracking each pixel touched by rendered primitives in a table by setting a flag in said table corresponding to each touched pixel;

suspending reads of pixel information if a pixel that will be touched by a primitive to be rendered is flagged; and

clearing said flags and resuming reads after pixel information of said flagged pixels has been updated or discarded.

13. The method of Claim 12, wherein said table comprises a first bit and a second bit and wherein said first bit flags all pixels whose information has not yet been updated or discarded and said second bit flags pixels which will be affected by said primitive to be rendered.

14. The method of Claim 12, wherein said flags corresponding to said pixels are indexed according to the xy coordinates of said pixels.

AD

15. (AMENDED) The method of Claim 12, wherein suspending and clearing steps are not performed if said primitive to be rendered appears in a history list of recently rendered primitive data.

16. A method of computer graphics memory management, comprising the actions of:

tracking each pixel touched by rendered primitives in a table comprised of a first bit entry and second bit entry for each pixel;

setting a flag in said second bit entry corresponding to each pixel which will be touched by a primitive to be rendered;

copying said flagged second bit entries to said first bit entries after each primitive is rendered;

suspending reads of pixel information if said first bit entry of a pixel that will be touched by a primitive to be rendered is flagged; and

clearing all flags and resuming reads after pixel information of all said flagged pixels has been updated or discarded.

17. The method of Claim 16, wherein a table is used to store said flags.

18. The method of Claim 16, wherein a table is used to store said flags, said table comprising a first bit and a second bit and wherein said first bit flags all pixels whose information has not yet been updated or discarded and said second bit flags pixels which will be affected by said primitive to be rendered.

19. The method of Claim 16, wherein said flags corresponding to said pixels are indexed according to the xy coordinates of said pixels.

20. The method of Claim 16, wherein suspending and clearing steps are not performed if said primitive to be rendered appears in a history list of recently rendered primitive data.

21. A method of computer graphics memory management, comprising the actions of:

- keeping a record of each pixel affected by a rendered primitive;
- removing a pixel from said record if said pixel information is updated or discarded; and
- suspending reads of pixel information if a primitive to be rendered will affect a pixel whose entry remains in said record.

22. The method of Claim 21, wherein said record is kept in a table.

23. The method of Claim 21, wherein said record is kept in a table, said table comprising a first bit and a second bit and wherein said first bit flags all pixels whose information has not yet been updated or discarded and said second bit flags pixels which will be affected by said primitive to be rendered.

24. The method of Claim 21, wherein said suspending step is not performed if said primitive to be rendered appears in a history list of recently rendered pixel data.

25. A method of computer graphics memory management, comprising the actions of:

- tracking each primitive to be rendered and the pixel information each said primitive will affect; and
- suspending reads of pixel information between rendering of each said primitive only if the primitive to be rendered will affect pixel information that has been previously rendered and has not had sufficient time to update.

26. The method of Claim 25, wherein a table is used to track said primitives.

27. The method of Claim 25, wherein a table is used to track said primitives, said table comprising a first bit and a second bit and wherein said first bit flags all pixels whose information has not yet been updated or discarded and said second bit flags pixels which will be affected by said primitive to be rendered.

28. The method of Claim 25, wherein said suspending step is not invoked if said primitive to be rendered appears in a history list of recently rendered primitive data.

29. A method of computer graphics memory management, comprising the actions of:

- assigning a unique identifier to each primitive before it is rendered;
- tracking the pixels affected by each rendered primitive using said unique identifier; and

- if a primitive to be rendered will affect pixel information that has been affected by a previously rendered primitive, suspending reading of primitive information unless said affected pixel information has been updated.

30. The method of Claim 29, wherein a table is used to track said pixels with said unique identifier.

31. The method of Claim 29, wherein reads are not suspended if said primitive to be rendered appears in a history list of recently rendered primitive data.

32. The method of Claim 29, wherein a table is used to track said pixels with said unique identifier, said table comprising a first bit and a second bit and wherein said first bit flags all pixels whose information has not yet been updated or discarded and said second bit flags pixels which will be affected by said primitive to be rendered.

33. A pipelined graphics system, comprising:

display hardware; and

video rendering hardware comprising, a read unit to read information regarding a primitive to be rendered to be displayed by said display hardware, a write unit to update modified information regarding primitives, and a read monitor unit to prevent primitive information which has been modified from being read by said read unit before said write unit has updated said information;

wherein said read monitor unit tracks pixel information affected by previously rendered primitives and suspends reads if a primitive to be rendered will affect pixel information which has been affected by a previously rendered primitive until said pixel information has had time to update.

34. The system of Claim 33, wherein said read monitor unit implements a table to track said pixel information.

35. The system of Claim 33, wherein said read monitor unit implements a table to track said pixel information, said table comprising a first bit and a second bit and wherein said first bit flags all pixels whose information has not yet been updated or discarded and said second bit flags pixels which will be affected by said primitive to be rendered.

36. The system of Claim 33, wherein said read monitor does not suspend reads if said primitive to be rendered appears in a history list of recently rendered primitive data.

37. A graphics processing subsystem, comprising:

at least four functionally distinct processing units, each including hardware elements which are customized to perform a rendering operation which is not performed by at least some others of said processing units; at least some ones of said processing units being connected to operate asynchronously to one another; and

a frame buffer, connected to be accessed by at least one of said processing units;

said processing units being mutually interconnected in a pipeline relationship, such that said processing units jointly provide a pipelined multiple-instruction-multiple-device (MIMD) graphics processing architecture;

wherein suspension of reads of pixel information during rendering of primitives is invoked only selectively.

---

38. (AMENDED) The subsystem of Claim 37, wherein a table is used to determine if reads should be suspended.

A3 39. (AMENDED) The subsystem of Claim 37, wherein a table is used to determine if reads should be suspended, said table comprising a first bit and a second bit and wherein said first bit flags all pixels whose information has not yet been updated or discarded and said second bit flags pixels which will be affected by said primitive to be rendered.

40. (AMENDED) The subsystem of Claim 37, wherein suspension of reads is invoked unless said primitive to be rendered appears in a history list of recently rendered primitive data.

---